

# „Roboterlader“ mit LEGO Mindstorms

Volodymyr Drovnikov, Elektrotechnik und Informationstechnik  
Otto-von-Guericke-Universität Magdeburg

*Jedes Jahr findet an der Otto-von-Guericke-Universität Magdeburg ein Design-Workshop statt, bei dem Studierende der Ingenieurwissenschaften einen Roboter aus LEGO entwerfen. Im Rahmen des diesjährigen Projekts habe ich einen besonderen Mechanismus geschaffen, ich bin „Roboterlader“. Dieses Projekt trägt dazu bei, das Risiko von Verletzungen und Schäden am Arbeitsplatz, insbesondere in Lagerhallen, zu verringern. Für die Gestaltung kamen LEGO-Mindstorms-Sets und der NXT-Steuerungscomputer zum Einsatz. Dieser Artikel beschreibt die Struktur und Funktionen des Mechanismus. Auch die bei der Gestaltung aufgetretenen Probleme und deren Lösungen wurden erwähnt.*

**Schlagwörter** — Hinderniserkennung, Ultraschall, Roboter, MATLAB, Programm.

## I. EINLEITUNG

Die rasanten Fortschritte in der Robotik eröffnen neue Möglichkeiten, um auf globale Herausforderungen zu reagieren – sei es in der Industrie, der Medizin oder der öffentlichen Sicherheit. Besonders eindrucksvoll zeigte sich dies nach der Katastrophe in Fukushima, als spezialisierte Roboter eingesetzt wurden, um die hochgradig verstrahlten Gebiete zu erkunden und Informationen zu sammeln (Abb. 1). Ohne ihr Eingreifen wären unzählige Menschenleben in Gefahr gewesen.

Angesichts der zunehmenden Risiken in Krisensituationen – von Naturkatastrophen bis hin zu industriellen Unfällen – ist die Weiterentwicklung dieser Technologie unerlässlich. Intelligente, vielseitige Roboter können nicht nur Effizienz und Sicherheit erhöhen, sondern auch das Risiko für Menschen in gefährlichen Umgebungen erheblich reduzieren.

Doch mit der steigenden Bedeutung der Robotik wächst auch die Verantwortung. Es reicht nicht aus, neue Technologien nur zu entwickeln – wir müssen sicherstellen, dass sie ethisch vertretbar und im Sinne der Gesellschaft eingesetzt werden. Eine klare Regulierung und durchdachte Strategien sind erforderlich, um die Potenziale der Robotik zu maximieren und gleichzeitig Risiken zu minimieren.

Die Zukunft gehört den Maschinen – doch es liegt in unseren Händen, sie klug und verantwortungsbewusst zu nutzen.

## II. VORBETRACHTUNGEN

In diesem Abschnitt werden die zentralen Werkzeuge dieses Projekts vorgestellt und ihre Funktionen im Zusammenhang mit der Durchführung spezifischer Aufgaben oder der Reaktion auf Eingaben beschrieben.

**Roboterprogrammierung.** Die Technologie basiert auf LEGO-Bausteinen, elektronischen Komponenten und spezieller

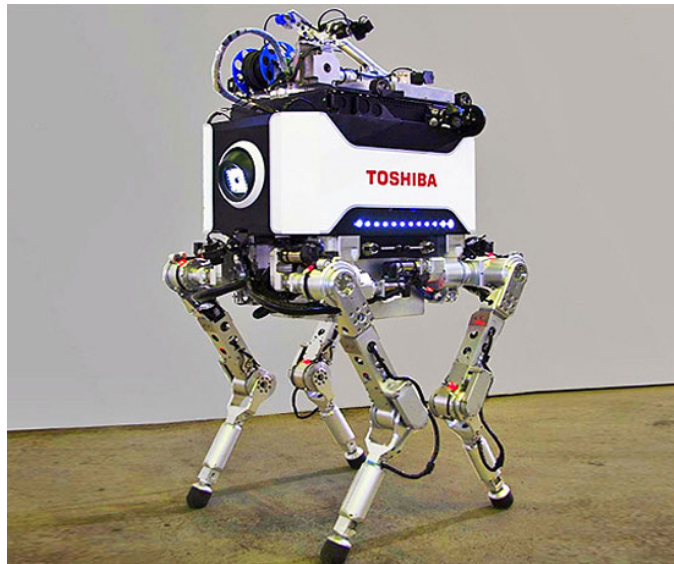


Abbildung 1: Tetrapode-Roboter in Fukushima [1]

Software zur Entwicklung und Steuerung von Robotern. Die aktuelle Version von NXT bietet eine breite Auswahl an Sensoren und Motoren sowie eine benutzerfreundliche grafische Programmierungsumgebung.

Im Rahmen des Otto-von-Guericke-Universitätsprojekts 2025 wurde ein Roboter auf Basis von NXT entwickelt und gebaut. Diese Plattform erleichtert die Umsetzung technischer Konzepte erheblich und beschleunigt den Entwicklungsprozess dank ihrer intuitiven und skalierbaren Struktur. Die neueste Version von Mindstorms NXT bietet eine erweiterte Funktionalität und noch höhere Benutzerfreundlichkeit im Vergleich zu ihrem Vorgänger. Dadurch eignet sie sich besonders für wissenschaftliche und technische Anwendungen, etwa die Entwicklung und Programmierung eines Manipulator-Roboters.

**MATLAB als Entwicklungsumgebung.** MATLAB (Matrix Laboratory) ist eine von MathWorks entwickelte Programmiersprache und Entwicklungsumgebung. Sie ermöglicht numerische Berechnungen, Datenvisualisierung, Simulationen und Analysen. MATLAB bietet leistungsstarke Werkzeuge für die Matrixmanipulation, Signal- und Bildverarbeitung sowie für mathematische und ingenieurwissenschaftliche Berechnungen.

In der Robotik kann MATLAB vielseitig eingesetzt werden – von der Entwicklung und Erprobung von Steuerungsalgorithmen über die Simulation der Roboterdynamik bis hin zur Sensordatenanalyse oder dem Entwurf eines

Überwachungssystems. Daher sind grundlegende Programmierkenntnisse in MATLAB besonders wertvoll für die erfolgreiche Umsetzung technischer Mechanismen.

### III. ROBOTERENTWICKLUNG

#### A. Aufbau

Die Konstruktion eines Roboters ist technisch simpel. Doch damit das System effektiv arbeiten kann, muss es nicht nur äußere Reize wahrnehmen, sondern auch auf diese reagieren und sich entsprechend bewegen. Dafür spielen Motoren und Sensoren eine entscheidende Rolle: Sie ermöglichen es dem Roboter, Lasten zu manövrieren, präzise Bewegungen auszuführen und auf verschiedene Umweltbedingungen zu reagieren. Im Folgenden werden die wichtigsten Sensoren und Motoren sowie ihre spezifischen Eigenschaften aufgeführt, die für eine optimale Funktionalität des Roboters erforderlich sind.

- *Motoren*

Für den Bau des Mechanismus werden drei große Motoren benötigt (Abb.2). Zwei davon steuern die Vorwärts- und Rückwärtsbewegung sowie das Abbiegen des Roboters, während der dritte Motor für das Anheben der Klaue verantwortlich ist. Diese Kombination ermöglicht eine präzise Steuerung und flexible Bewegungen, wodurch der Roboter effizient auf verschiedene Aufgaben reagieren kann.



Abbildung 2: großer Motor [2]

- *Farben Sensor*

Um eine präzise Steuerung zu gewährleisten, sind zwei Farbsensoren (Abb. 3) symmetrisch an der Unterseite des Roboters angebracht. Ihre Hauptaufgabe besteht darin, die Farbe Schwarz zu erkennen und somit die Position des Roboters auf der Strecke zu überwachen. Sobald einer der Sensoren eine schwarze Linie erfasst, passt der Roboter automatisch seine Bewegung an, um auf Kurs zu bleiben. Dieses intelligente Steuerungssystem ermöglicht eine zuverlässige Navigation und verhindert ungewollte Abweichungen.

- *Ultraschallsensor*

Ein moderner Roboter muss nicht nur Objekte erkennen, sondern auch intelligent darauf reagieren. Dafür kommt ein Ultraschallsensor (Abb.4) zum Einsatz, der die Distanz zu Boxen misst und ihre Position bestimmt. Nach der ersten Messung erfasst der Sensor eine zweite Distanz, vergleicht beide Werte

und entscheidet, ob eine Anpassung notwendig ist. Falls die Höhenunterschiede ausgeglichen werden müssen, greift die Klaue ein und positioniert die Box präzise auf das gewünschte Niveau. Diese Technologie ermöglicht eine effiziente und automatisierte Objekthandhabung, die besonders in logistischen und industriellen Anwendungen von großem Vorteil ist.



Abbildung 3: Farbsensor [2]

#### B. PROGRAMMIERUNG

Die Verbindung von moderner Programmierung und innovativer Technik spielt eine entscheidende Rolle in der Robotik. Der NXT-Controller fungiert als Steuerzentrale eines Roboters und ermöglicht die präzise Ansteuerung von Motoren und Sensoren über verschiedene Schnittstellen, darunter USB. Gesteuert wird das System über MATLAB, eine leistungsstarke Plattform für numerische Berechnungen und automatisierte Prozesse.



Abbildung 4: Ultraschallsensor [2]

Mit dem NXT-Toolkit lassen sich Sensordaten erfassen und Motorbewegungen exakt steuern. Eine klare Kennzeichnung der Anschlüsse – Buchstaben für Motoren, Zahlen für Sensoren – erleichtert dabei die Konfiguration. Die NXT-Toolbox stellt eine Vielzahl von Befehlen zur Verfügung, die eine intuitive Programmierung ermöglichen: So steuert der Befehl „*motorA.setPower(-30)*“ die Motorleistung, während „*SensorUltrasonic.getDistance()*“ die Distanzmessung mit dem Ultraschallsensor durchführt.

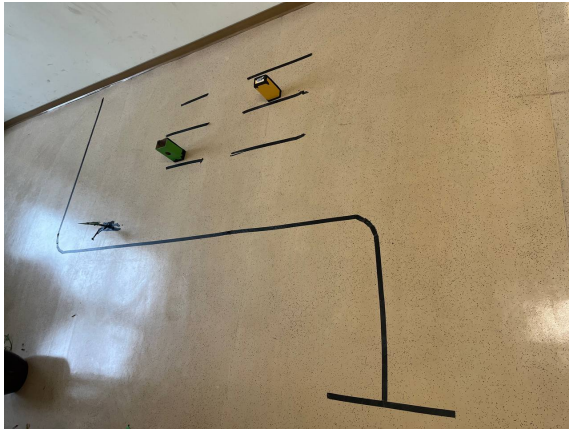


Abbildung 5: Fahrfeld des „Roboterladers“

MATLAB bietet neben klassischen Programmierstrukturen wie Schleifen und Bedingungen eine direkte Schnittstelle zur Hardware. Der NXT-Block empfängt Befehle vom Computer und überträgt sie an den Roboter, der daraufhin die gewünschten Aktionen ausführt. Durch einfaches Drücken der „Run“-Taste startet der gesamte Prozess – eine effiziente Lösung für den schnellen Einsatz in technischen und wissenschaftlichen Projekten.

```
COM_CloseNXT all;
h = COM_OpenNXT();
COM_SetDefaultNXT(h);
OpenLight(SENSOR_1, 'ACTIVE');
OpenLight(SENSOR_2, 'ACTIVE');
motorLeft = MOTOR_B;
motorRight = MOTOR_C;
while true
    leftValue = GetLight(SENSOR_1);
    rightValue = GetLight(SENSOR_2);

    if leftValue >= 500 && leftValue <= 600
        NXT_SetOutputState(motorLeft, -30, 0, 'Brake', 0, 0, 0);
        NXT_SetOutputState(motorRight, 30, 0, 'Brake', 0, 0, 0);
    else
        NXT_SetOutputState(motorLeft, 30, 0, 'Brake', 0, 0, 0);
        NXT_SetOutputState(motorRight, 30, 0, 'Brake', 0, 0, 0);
    end
    pause(0.1);
end
CloseSensor(SENSOR_1);
CloseSensor(SENSOR_2);
COM_CloseNXT(h);
```

Abbildung 6: Codekette

In (Abb. 6) wird ein Beispielcode präsentiert, der zur Anpassung des Roboters dient, wenn der linke Sensor mit einer schwarzen Linie in Kontakt kommt.

### C. PROBLEME

Während der Entwicklung des Roboters wurden viele Probleme festgestellt. Eines der interessantesten soll hier hervorgehoben werden: Das erste Problem bestand darin, dass ein EV3-

Lichtsensoren an den NXT angeschlossen wurde, was dazu führte, dass falsche Werte geliefert wurden. Dieses Problem konnte jedoch leicht behoben werden, indem ein für den NXT geeigneter Sensor verwendet und ausgetauscht wurde.

Als der Roboter bereits in der Lage war, der schwarzen Linie zu folgen (Abb. 5), stellte sich die Herausforderung, zwei Kästchen zu erfassen, ohne unnötige Werte in die Variablen einzutragen. Ursprünglich wurde die Idee verfolgt, eine fahnenförmige Identifikationsmarkierung anzubringen, um die Entfernung festzulegen, bis zu der ein Codeblock ausgeführt werden sollte, um die Position der Kästchen zu bestimmen. Jedoch wurde festgestellt, dass der Roboter nicht erkannte, dass dasselbe Kästchen mehrfach erfasst wurde. Aus diesem Grund wurde die Idee der Markierung verworfen.

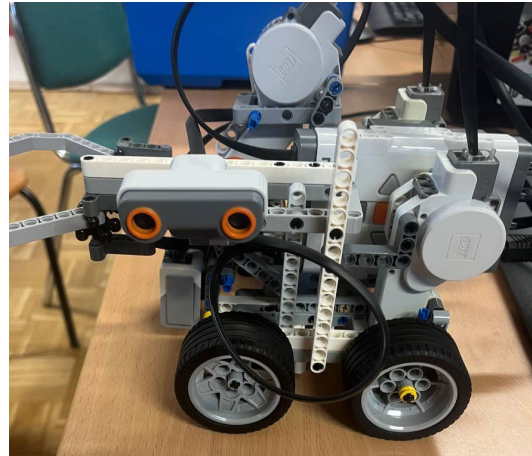


Abbildung 7: fertiger „Roboterlader“

Dieses Problem wurde gelöst, indem die Entfernung nur dann gemessen wurde, wenn sie unter einem bestimmten Wert lag. Die Messung wurde dann gestoppt, bis der Roboter den Raum zwischen den Kästchen erreichte und erneut einen Wert über dem in der Bedingung definierten Schwellenwert registrierte.

Auch bei den Farbsensoren traten Probleme auf, da der verwendete Sensortyp den Farberkennungsmodus nicht unterstützte. Daher wurde eine Analyse der Lichtreflexionen verschiedener Farben durchgeführt, um diese zur Bestimmung der jeweiligen Farbe heranzuziehen.

## IV. ERGEBNISDISKUSSION

Die Erreichung des gesetzten Ziels wurde bestätigt. Der Mechanismus (Abb. 7) funktioniert fehlerfrei; die Sensoren und Motoren arbeiten synchron und präzise gemäß der Befehlssequenz. Ein Mechanismus, der eine solche zuverlässige Leistung zeigt, besitzt das Potenzial für eine erfolgreiche Integration in verschiedene Umgebungen und reale Anwendungen. Das Ziel des Projekts wurde erfolgreich erreicht.

## LITERATURVERZEICHNIS

- [1] *Liquidationsroboter in Fukushima*  
<https://www.zeit.de/wissen/2011-04/roboter-katastrophen>
- [2] *LEGO NXT*

<https://www.lego.com/cdn/product-assets/product.bi.core.pdf/4520734.pdf>

[3] **WIKIPEDIA: LEGO NXT**

[https://de.wikipedia.org/wiki/Lego\\_Mindstorms\\_NXT](https://de.wikipedia.org/wiki/Lego_Mindstorms_NXT)

[4] **SOFTWARE LEGO NXT**

[https://www.lego.com/de-de/service/help/mindstorms\\_nxt/mindstorms\\_nxt/lego-mindstorms-nxt-software-downloads-kA009000001dck4CAA](https://www.lego.com/de-de/service/help/mindstorms_nxt/mindstorms_nxt/lego-mindstorms-nxt-software-downloads-kA009000001dck4CAA)

[5] **Projekt Instagram:**

<https://www.instagram.com/legomindstormsgruppe15wise2025?>